REMARKS

Claims 37-44, 46-50, 52-58, and 60 were pending and rejected. Claims 1-36, 45, 51 and 59 were previously cancelled. In response, Applicants offer to amend claims 37, 38, 47, 57 and 60 as set forth above to address the formality and statutory subject matter issues. No new matter has been introduced. Accordingly, claims 37-44, 46-50, 52-58, and 60 remain pending, and their scopes unchanged. Thus, no new search is required. For at least the reasons set forth below, Applicants respectfully submit that the claims are in condition of allowance. Therefore, entry of the offered amendments and allowance of the remaining pending claims are respectfully requested.

Claims Objections

Claims 37, 38, 46, 50, 52-55 and 57 were objected to for various formalities issue. Claims 37, 38, 50, 52-55 and 57 have been amended to overcome the Examiner's objections. In particular, the "prior to …" phrase of claim 37 has been amended to conform exactly as the Examiner's interpretation. The same phrase in independent claims 47, 57 and 60 has also been likewise amended (even though they were not objected to by the Examiner). Additionally, Applicants hereby confirm that indeed the Examiner's interpretation of the letter "s" of the word "frames" was to be deleted is correct. Claim 46 is now so shown, reflecting the entry of the amendment through the last response. Accordingly, all formality issues have been addressed. Withdrawal of the objections is respectfully requested.

35 USC §101 Rejections

In the subject Office Action, claims 47-50 and 52-55 were rejected under 35 USC §101. In response, Applicants have amended independent claim 47 to clearly recite "non-transitory computer-readable storage medium," and dependent claims 50 and 52-55 to clearly recite "the computer-readable storage medium," as recommended by the Examiner, overcoming the rejections. Thus, withdrawal of the rejection is respectfully requested.

35 USC §103 Rejections

In the subject Final Office Action, claims 37-39, 43-44, 46-48, 50, 52-58, and 60 were rejected under 35 USC §103(a) as being unpatentable over Requa ("The Piecewise Data Flow Architecture: Architectural Concepts,") in view of Patterson et al. ("Computer Architecture, A quantitative Approach.). In response, Applicants respectfully traverse the Examiner's rejections.

In the Final Office Action, the Examiner maintained the rejections. The Examiner reasoned that except for "teaching a buffer for each node, receiving instructions prior to the operands for said instruction being present," Requa teaches all recitations of the independent claims. The deficiencies of Requa are remedied by Patterson. Applicants respectfully traverse both of these positions.

Claim 1 recites in pertinent part:

"assigning a group of instructions selected from a plurality of groups of instructions partitioned from a program, to a subset of interconnected computation nodes preselected from a plurality of interconnected computation nodes, the instructions having respective associated operands, and each computation node includes a store and an execution unit having one or more arithmetic logic units, floating point units, memory address units, or branch units;

loading a subset of instructions of the assigned group of instructions into a frame of buffers comprising the stores disposed on the preselected subset of interconnected computation nodes having been assigned the group of instructions, wherein the loading is performed prior to availability of the associated operands of the subset of instructions …"

In the last response, Applicants argued that these two recitations are not taught by Requa and not remedied by Patterson. The reason is because claim 37, not only recites the performance of an "assigning …" operation, and a "loading …" operation, but recites the two operations with a particular relationship and in a particular context. More specifically, claim 37 recites the performance of the "loading …" operation is on "a subset of instruction of the ... group of instructions," that is the object of the "assigning …" operation. Further, the "subset of instructions" is loaded into the "stores" of the assigned "computation nodes," "prior to availability of the respective associated operands of the subset of instructions."

In contrast, Requa merely disclosed a data flow architecture that includes a Block Processing section, an Instruction Issue section and an Instruction Processing section having a

number of scalar processors, a memory processor and a SIMD processor. As described in the section cited by the Examiner, under Requa "Instructions are grouped (by the Block Processing section) into relatively small blocks" and dispatched on a block by block basis to the Instruction Issue section, which in turn issues the instructions to one or more of the scalar processors, the memory processor or the SIMD processor to execute, as the operands of the instructions become available.

Under Requa, there is no concept that a block of instructions is overtly "assigned" to be executed by a subset of the scalar, memory and SIMD processors, when the block instructions is dispatched from the Block Processing section to the Instruction Issue section. Which processor is ultimately employed to execute an instruction, and thus the combination of processors employed to execute a block of instructions, is decided by the Instruction Issue section (based on the instruction type). Accordingly, even if we ignore the fact that Requa's scalar, memory and SIMD processors are not "computation nodes" (as each of the scalar, memory and SIMD processors does not include "… a store …" to be collectively employed to hold the "loaded" instructions "prior to availability of the respective associated operands…"), Requa still fails to teach or suggest the recitation of "assigning a group of instructions selected from a plurality of groups of instructions partitioned from a program, to a subset of interconnected computation nodes preselected from a plurality of interconnected computation nodes, …" of claim 37.

It follows then Requa fails to teach or suggest the "loading …" operation, as the "loading … operation" is to be performed on "a subset of instructions" of the assigned "group of instructions," object of the "assigning …" operation."

In response to Applicant's arguments, at paragraph 36, the Examiner separately responded to three aspects of Applicants arguments, the argument with respect to lack of teachings of the recited "assigning …" and "loading …" operations, and the associated arguments of "computational node." The Examiner made various statements, such as "if the instructions make it to a processor, then clearly, they were assigned ..," "since each processor cannot hold simultaneously every single instruction, … only a subset is loaded," and "unreasonable" for Applicants to argue the "processors" are not "computation nodes."

Applicants respectively disagree with the Examiner, and further digress on Applicants' arguments. Claim 37, as discussed above, recites an "assigning …" and a "loading …"

operation. The two recitations did not just say "loading instructions into an execution unit and executing the instructions." Had that been the recitations, the Examiner's response may be reasonable. In claim 37, as discussed earlier, the "assigning …" and "loading …" operations are recited with relationship to each other, qualified with language that recites the objects and contexts of the operations. For the Examiner to meet the burden of establishing the prima facie cases of obviousness, it is necessary for the Examiner to identify the disclosure of the references that teaches or suggests each and every aspect of these two recited operations.

As discussed earlier, in claim 37, the "assigning …" and "loading …" operations are recited with particular relationship in that the "loading …" operation is performed on "a subset of the instructions" assigned to be executed by a subset of the available "computation nodes" preselected from a plurality of "computational loads." The "assigning …" is an affirmatively performed operation. Further, the "computation nodes" include "stores" that collective hold the loaded subset of instructions assigned to the subset of computation nodes, "prior to availability of the respective associated operands of the subset of instructions."

In Requa, the scalar, memory and SIMD processors are disclosed as employed to respectively execute scalar, memory and SIMD instructions. When a block of instructions is "loaded …" for execution by these processors, they are loaded into the Instruction FIFO of the Instruction Issue section. The Instruction FIFO is NOT distributedly disposed in the scalar, memory and SIMD processors, as required by claim 37. Without the "stores," the scalar, memory and SIMD cannot be read as equivalent to the recited "computational nodes" of the "loading …" operation. Thus, it follows, the "loading (of) a subset of instructions of the assigned group of instructions into a frame of buffers comprising the stores disposed on the preselected subset of interconnected computation nodes … (for execution) … prior to availability of the associated operands of the subset of instructions," cannot be said as having been taught by Requa's loading of a block of instructions into the Instruction FIFO.

In Requa, whether an instruction is executed by the scalar, the memory or the SIMD processor, is strictly dependent on the type of the instructions, i.e.., whether it is a scalar, memory or SIMD instruction. Requa does not have any notion of affirmatively performing an "assigning …" operation as recited, i.e. "assigning a group of instructions selected from a plurality of groups of instructions partitioned from a program, to a subset of interconnected

computation nodes preselected from a plurality of interconnected computation nodes," where "each computation node includes a store and an execution unit …," where the "stores" form "a frame of buffers" for "loading (the) the subset of instructions …," where the "the loading is performed prior to availability of the associated operands of the subset of instructions."

Accordingly, Applicants respectfully submit the recited "assigning …" and "loading …" operations, with the recited relationship, as well as the recited objects and context of operations, when viewed as a whole, as required by law, are not taught nor suggested by Requa. Thus, even if we assume the Examiner's reading of Patterson is correct, the combination still fails to teach or suggests all recitations of claim 37.

With respect to the Examiner's reading of Patterson, Applicants respectfully traverse the Examiner's assertion that "a reservation station system such as Patterson allows each processor to directly send data to another processor via the bus, without the data first being stored in the register file." Applicants respectfully directs the Examiner attention to Figure 4.8 on page 253, where it is clearly illustrated that the FP adders and FP multipliers output onto the common data bus (CDB) (bottom of the Figure), and the CDB (running along the right edge of the Figure) "dumps" into the FP registers (top right corner of Figure), the store buffers (below the FP register) and the reservation stations (lower portion of the Figure) of the same processor. There is NO teaching of any output path that allows execution results to be directly provided to another processor.

Accordingly, for at least the foregoing reasons, Applicants submit amended claim 37 is patentable over Requa and Patterson, individually or in combination, 35 USC §103(a).

Claims 47, 57 and 60 have been similarly amended as claim 37. Thus, for at least similar reasons, claims 47, 57 and 60 are patentable over the cited references.

Claims 38-44, 46, 48-50, 52-56, and 58 depend from either claim 37, 47 and 57, incorporating their recitations. Therefore, for at least similar reasons, claims 38-44, 46, 48-50, 52-56, and 58 are patentable over the cited references. Claims 38-44, 46, 48-50, 52-56, and 58 are further patentable over the cited references by virtue of their additional recitations.

Conclusion

In view of the foregoing, Applicant submits that, all remaining pending claims are now in condition of allowance. Therefore, early issuance of a Notice of Allowance is respectfully requested. Should there be any question with Applicants' response, Applicants invite the Examiner to contact the undersigned at 206-381-8819 (Direct).

Lastly, the Commissioner is hereby authorized to charge shortages or credit overpayments to Deposit Account No. 500393.

Respectfully submitted,
Schwabe, Williamson & Wyatt, P.C.

Dated: September 22, 2010

/Al AuYeung/
Al AuYeung
Reg. No. 35,432